# **Grazioso Salvare Dashboard README**

This project involves the creation of a dashboard for Grazioso Salvare, a rescue animal operations company. The dashboard allows users to filter, visualize, and map data related to rescue animals, helping them to make informed decisions.

## **Project Functionality**

The dashboard provides the following features:

* **Filtering Data**: Users can filter animal data based on specific rescue operations like Water Rescue, Mountain or Wilderness Rescue, and Disaster or Individual Tracking.
* **Data Table**: A dynamic data table displays the filtered results, allowing users to sort and select specific entries.
* **Visualization**: A pie chart visualizes the distribution of breeds for the filtered data.
* **Mapping**: A map shows the location of selected animals, providing a visual representation of where each animal is located.

## **Tools Used**

### **1. MongoDB**

* **Reason for Use**: MongoDB was chosen as the database due to its flexibility in handling unstructured data and its ability to easily interface with Python using the pymongo library.
* **Capabilities**: MongoDB provides powerful querying capabilities and supports complex data models, making it ideal for managing the animal shelter data.

### **2. Dash Framework**

* **Reason for Use**: Dash was selected for its ability to create interactive web applications directly from Python code. It seamlessly integrates with Plotly for visualizations and provides a robust structure for building web-based data dashboards.
* **Capabilities**: Dash allows for the creation of interactive and real-time dashboards, enabling users to filter data and see visualizations and maps update dynamically.

### **3. Dash Leaflet**

* **Reason for Use**: Dash Leaflet was used to provide mapping capabilities within the dashboard. It integrates well with Dash and allows for easy visualization of geospatial data.
* **Capabilities**: Dash Leaflet enables the display of dynamic maps with markers, tooltips, and popups, making it an excellent tool for geographic visualization.

## **Steps Taken to Complete the Project**

1. **Set Up MongoDB Connection**: The project began with establishing a connection to the MongoDB database, where the animal data is stored.
2. **Data Handling**: The data was retrieved from MongoDB and processed using Pandas to prepare it for display in the dashboard.
3. **Dashboard Development**: Using the Dash framework, the dashboard layout was designed, incorporating a data table, pie chart, and map.
4. **Interactive Features**: Callbacks were implemented to enable dynamic filtering of data and updating of the visualizations and map based on user input.
5. **Testing**: The dashboard was thoroughly tested to ensure that all functionalities worked as expected, including filtering, visualization, and mapping.

## **Challenges Encountered**

* **Handling ObjectID in MongoDB**: One challenge was dealing with the ObjectID returned by MongoDB, which caused issues with the Dash data table. This was resolved by removing the \_id field from the data before rendering it in the table.
* **Dynamic Updating of Components**: Ensuring that the dashboard components (e.g., data table, pie chart, map) updated dynamically based on user input required careful implementation of Dash callbacks. The challenge was overcome by debugging the callback functions and testing various scenarios.

**Screenshots**

